

VISTA EXTENSIBLE MARKUP LANGUAGE (XML) PARSER

Release Notes for Patch XT*7.3*58, Version 1.1

Changes

- Fixed undefined variable error
- Fixed incorrect handling of whitespace normalization within attribute values
- Extensively tested against Jim Clark's test suite of XML documents and fixed several parsing bugs
- Parser now preserves state of Kernel I/O variables
- Added US-ASCII as a supported character encoding
- Added MXMLCANO routine to generate canonical form of documents for testing purposes
- Inverted the meaning of the "V" flag in the OPTION parameter of the EN^MXMLPRSE entry point to make it more consistent with the meaning of the "W" flag. Now, the presence of the "V" flag suppresses validation of the document. Thus, by default the compiler validates a document.

Known Issues

- Only character encodings that contain the ASCII subset are supported
- Because the Kernel function FTG^%ZISH accesses files in text mode, certain control characters (which are disallowed by XML anyway) are stripped from the input stream and not detected by the parser. The effect is that the parser will not flag such documents as non-conforming. A secondary effect is the inability of the parser to distinguish a document ending with CR-LF sequence from one that does not.
- FTG^%ZISH opens files with a time-out parameter. The effect is that an attempt to reference a nonexistent external entity results in a delay of a few seconds before the error condition is signaled.
- The parser does not support URL's in entity system identifiers that specify special transport protocols (like HTTP or FTP). Currently only UNC style file references are supported (i.e., path + filename).
- The parser still allows external entities to contain substitution text that in some cases would violate XML rules that state that a document must be conforming in the absence of resolving such references. In other words, XML states that a non-validating parser should be able to verify that a document is conforming without processing external entities. This restriction constrains how token streams can be continued across entities. The parser recognizes most, but not all, of these restrictions. The effect is that the parser is more lax in allowing certain kinds of entity substitutions.
- There are many XML documents in use that are strictly non-conforming, yet many parsers will not reject them. Most commonly this involves the absence of whitespace designated as required by the XML specification. Currently, this parser enforces required whitespace even when the absence of such does not introduce syntactic ambiguity. The effect is that this parser may reject some documents that may be accepted by other parsers.